

# **METHOD, SYSTEM AND ARTICLE FOR DETECTING CRITICAL MEMORY LEAKS CAUSING OUT-OF-MEMORY ERRORS IN JAVA SOFTWARE**

## **ABSTRACT**

[0052] An analysis tool for specifically identifying the cause and location of critical memory leaks within a large software system is provided. The analysis tool transparently adds multiple processing threads that collaborate to identify sources of potential memory leaks in the software being analyzed. These threads provide notifications of object classes executing in the software that have had excessive instance counts or have exceeded the average lifetime of an object. A stack walkback is performed on objects being monitored as an out-of-memory condition approaches and while there is still sufficient memory available to unwind and store the stack walkback. By saving the stack walkbacks just prior to the out-of-memory failure, the analysis tool operator is thereby provided sufficient information to make a final determination of objects that are possible candidates for memory leaks and subsequently modify the software to reduce or eliminate the memory leaks.